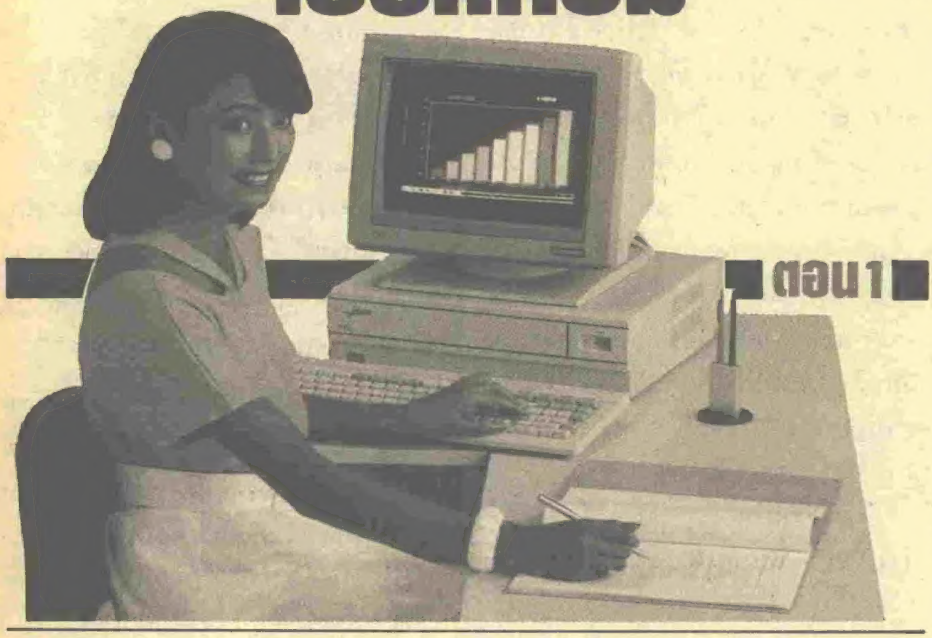
(13)

เทคนิคการออกแบบโปรแกรม



ตอน 1

อาตาปี & VIP

คำนำ

หัวใจของงานวิศวกรรมสำคัญๆ ทั้งหลายก็คือการออกแบบ เมื่อเราต้องการสร้างบ้านสักหลังหนึ่ง เราไม่อาจจะเริ่มด้วยการขุดหลุมฝังเสา เอาคานมาตอกตะปู เอากระเบื้องมุงหลังคาได้ในทันทีทันใด แต่เราต้องเริ่มด้วยการออกแบบบ้านหลังนั้นก่อนว่า จะเอาเสาไว้ตรงไหน จะใช้คานขนาดเท่าใด ตรงไหนจะเป็นห้องรับแขก ห้องน้ำ ห้องครัว ฯลฯ การออกแบบบ้านโดยเขียนเป็นโครงร่างบนกระดาษนั้นยังดีตรงที่เราสามารถพิจารณาแบบบ้านนั้นได้ก่อนว่าถูกใจหรือไม่โดยยังไม่ต้องลงมือสร้างจริงๆ ถ้าหากไม่ถูกใจก็ลบทิ้งและเขียนขึ้นมาดูใหม่ ถ้าไม่มีแบบบ้านมาดูก่อน แต่ลงมือสร้างบ้านไปเรื่อยๆ เราอาจจะลงเอยด้วยสิ่งก่อสร้างที่ไม่สามารถคุ้มแดดคุ้มฝนตามที่ต้องการได้

งานพัฒนาโปรแกรมก็เช่นกัน โปรแกรมที่สำคัญๆ จำเป็นจะต้องได้รับการออกแบบ และนำมาพิจารณากันอย่างถี่ถ้วนก่อนที่จะลงมือเขียนเป็นโปรแกรม ถ้าหากด่วนเขียนเป็นโปรแกรมทีเดียว อาจได้ผลลัพธ์เป็นโปรแกรมที่ไม่สอดคล้องกับข้อกำหนดและความต้องการก็ได้

นักคอมพิวเตอร์ชาวไทยจำนวนมากมีความสามารถสูง สามารถพัฒนาโปรแกรมที่ซับซ้อนขึ้นใช้เองได้ แต่เมื่อกล่าวถึงการออกแบบโปรแกรม กลับมีเพียงไม่กี่คนเท่านั้นที่สนใจปฏิบัติ ที่เป็นเช่นนี้เพราะนักโปรแกรมชาวไทยส่วนมากอาศัยความชำนาญและประสบการณ์เป็นหลัก เวลาพัฒนาโปรแกรม นักโปรแกรมสามารถรวบรวมความคิดสร้างจินตนาการของโครงสร้างโปรแกรมไว้ในใจ และเขียนเป็นคำสั่งคอมพิวเตอร์ออกมาได้ทันทีทันใด

วิธีดังกล่าวนี้ไม่ใช่วิธีที่ผิด ถ้าผิดเราก็คงไม่สามารถพัฒนางานคอมพิวเตอร์ถึงระดับปัจจุบันได้ แต่วิธีนี้อิงอาศัยประสบการณ์มากเกินไป และไม่สามารถสอนหรือถ่ายทอดให้กันได้ ถ้าหากงานที่ต้องเขียนโปรแกรมเป็นงานที่ซับซ้อนมากๆ วิธีดังกล่าวก็อาจใช้ไม่ได้ผล หรือใช้ได้แต่โปรแกรมที่ได้อาจมีข้อผิดพลาดมาก ปัจจุบันวงการคอมพิวเตอร์พยายามหาทางทำให้การใช้คอมพิวเตอร์ในด้านต่างๆ เช่น การเขียนโปรแกรม และการออกแบบฐานข้อมูลมีความเป็นวิทยาศาสตร์มากขึ้น และพึ่งพิงอาศัยประสบการณ์ส่วนตัวน้อยลง ดังนั้น จึงมีผู้หันมาสนใจเทคนิคการออกแบบโปรแกรมกันมากขึ้น เพราะได้รับการพิสูจน์แล้วว่า เป็นวิธีที่เป็นระบบและใช้ได้ผลดี

ความสำคัญอีกประการหนึ่งของการออกแบบโปรแกรมอยู่ที่ว่า โปรแกรมทั้งหลายนั้นเมื่อเขียนเสร็จแล้วจะต้องมีผู้อ่านโปรแกรมสองพวก พวกหนึ่งก็คือตัวคอมพิวเตอร์ซึ่งจะต้องอ่านและทำงานตามคำสั่งในโปรแกรม ส่วนอีกพวกหนึ่งก็คือนักโปรแกรมที่อยู่ในทีมงานพัฒนาโปรแกรมหรือที่จะปรับปรุงแก้ไขโปรแกรมต่อไปภายหน้า ซึ่งนักโปรแกรมเหล่านี้จำเป็นจะต้องเข้าใจว่าโปรแกรมทำงานอะไร และทำอย่างไรด้วย วิธีหนึ่งที่จะทำให้นักโปรแกรมเหล่านี้เข้าใจโปรแกรม และการทำงานของโปรแกรมได้ดีขึ้นก็โดยการจัดทำเอกสารเกี่ยวกับการออกแบบโปรแกรม แล้วจัดประชุมพิจารณาศึกษาเอกสารนั้นให้มีความเข้าใจตรงกันทั้งทีม

การจัดทำเอกสารการออกแบบโปรแกรมสำหรับใช้ในระหว่างการพัฒนาโปรแกรมนั้นเป็นงานที่ต้องใช้เวลาและแรงงานมาก ดังนั้น **วิธีการออกแบบ**

โปรแกรมสมัยใหม่จึงเน้นที่การจัดทำเอกสารให้รัดกุม กระชับ และมีมาตรฐานให้ทุกคนเข้าใจได้ตรงกัน นอกจากนั้นวิธีการสมัยใหม่บางอย่างยังนำคอมพิวเตอร์มาช่วยในการออกแบบโปรแกรมด้วย

จุดประสงค์อีกประการหนึ่งของการออกแบบก็เพื่อให้โปรแกรมที่เป็นผลลัพธ์ของงานพัฒนาระบบนั้นมีความถูกต้องสมบูรณ์ และมีที่ผิดพลาดน้อยที่สุด ในงานพัฒนาโปรแกรมที่สำคัญๆ บางอย่างนั้น จุดประสงค์แค่นี้ยังไม่พอ ทว่าต้องมีการใช้เทคนิคในการควบคุมตรวจสอบการทำงานของตัวเองว่าจะไม่มีการเกิดความผิดพลาดขึ้นด้วย

ในบทความนี้เราจะได้ศึกษาลักษณะทั่วไป และปรัชญาของการออกแบบโปรแกรมอย่างกว้างๆ ส่วนรายละเอียดของวิธีการออกแบบโปรแกรมแต่ละวิธีนั้นจะได้กล่าวโดยละเอียดต่อไป

ลักษณะของกระบวนการออกแบบ

การออกแบบสิ่งต่างๆ ไม่ว่าจะเป็นอาคาร เสื้อผ้า หรือโปรแกรมนั้น เป็นงานที่ขึ้นอยู่กับเฉพาะตัวบุคคล และต้องมีการคิดย้อนกลับไปกลับมาหลายเที่ยว โดยนักออกแบบจะต้องใช้ความรู้และประสบการณ์ ตลอดจนแรงดลใจในการสร้างสรรค์แนวทางขั้นต้นขึ้นมาก่อนว่า อาคารนั้นควรมีลักษณะอย่างไร เสื้อผ้าควรมีรูปแบบอย่างไร หรือโปรแกรมควรมีลักษณะอย่างไร จากนั้นเขาก็ใช้เครื่องมือสำหรับออกแบบมาวิเคราะห์แนวทางเหล่านี้ว่า มันสอดคล้องกับความต้องการและข้อกำหนดหรือไม่ ถ้าไม่สอดคล้องก็ต้องนึกหาแนวทางใหม่ เท่าที่กล่าวมานี้เป็นเพียงวิธีการคร่าวๆ ซึ่งการออกแบบจริงๆจะซับซ้อนกว่านี้มาก

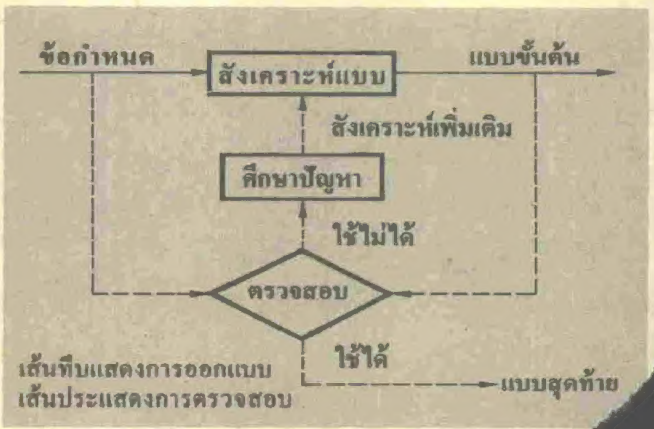
ยกตัวอย่างเช่น เรามักจะต้องพิจารณาแนวทางหรือ 'แบบ' ต่างๆ ที่เราคิดว่า แนวทางใด หรือแบบใด ที่ประหยัดและคุ้มค่ามากที่สุด ดังนั้นเราจึงจำเป็นจะต้องมีเครื่องมือวัด ซึ่งอาจจะเป็นหลักการ ทฤษฎี หรือแม้แต่ประสบการณ์ที่สามารถบอกได้ว่าแบบที่สองดีกว่าแบบที่หนึ่ง หรือกลับกัน เครื่องมือวัดเช่นนี้สามารถช่วยให้เราตัดสินใจเลิกพิจารณาแบบใดๆได้ตั้งแต่ในระยะแรกของการออกแบบถ้าหากเราเห็นว่ามันไม่น่าจะใช้การได้

เรื่องที่สำคัญที่สุดในการออกแบบก็คือ การใช้เทคนิคและวิธีการมาตรฐานที่ยอมรับกันทั่วไปในหน่วยงานนั้นหรือในวงกว้างกว่านั้น ถ้าหากจะใช้โปรแกรมนั้นในวงกว้าง การใช้มาตรฐานเป็นการบังคับไม่ให้ผู้ออกแบบเลือกใช้แต่แนวทางเฉพาะที่ตนเองคิดขึ้น และอาจไม่มีผู้อื่นเข้าใจ เทคนิคหรือวิธีพิเศษอาจสร้างปัญหาในภายหลังได้ถ้าหากเจ้าของเทคนิคพ้นหน้าที่หรือออกไปจากหน่วยงานแล้ว

กระบวนการสังเคราะห์ วิเคราะห์และลองผิดลองถูก

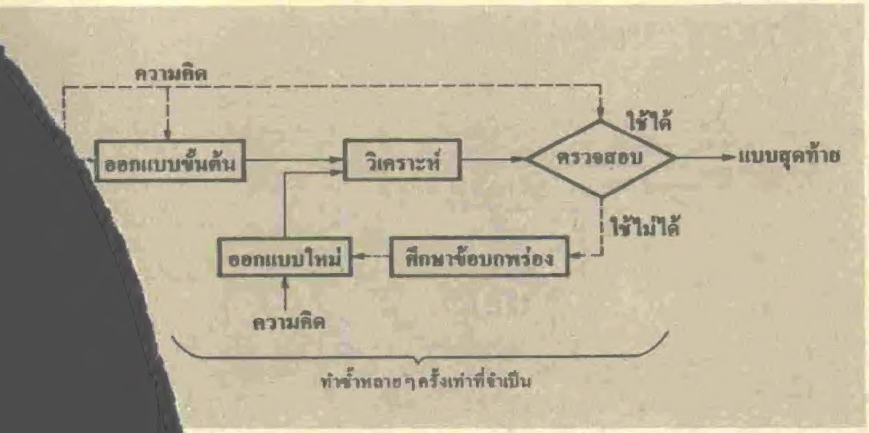
การออกแบบนั้นมีวิธีการหลายอย่าง เช่น การสังเคราะห์ การวิเคราะห์ และการลองผิดลองถูก เป็นต้น ไม่ว่าจะใช้วิธีใด การออกแบบจะเริ่มต้นจากข้อกำหนดหรือ specification ซึ่งระบุว่า ต้องการให้โปรแกรมทำอะไรบ้าง และได้ผลอย่างไร เมื่อการออกแบบสำเร็จแล้ว ผลที่ได้ก็จะต้องสอดคล้องตรงกับข้อกำหนดที่ตั้งเอาไว้ในตอนแรก บางครั้งในระหว่างการออกแบบเราสามารถพิสูจน์ได้ว่า ข้อกำหนดที่ตั้งเอาไว้นั้นเป็นไปไม่ได้ในทางปฏิบัติ หรือยังไม่มีความรู้ทางเทคโนโลยีใดในตอนนั้นที่จะทำให้สอดคล้องกับข้อกำหนดได้ หรือไม่มีใครที่มีความรู้พอที่จะออกแบบให้ตรงกับข้อกำหนดได้ ดังนั้น ก็จำเป็นจะต้องเปลี่ยนแปลงข้อกำหนดเดิม แล้วเริ่มต้นงานออกแบบกันใหม่

ถ้าหากข้อกำหนดที่มีอยู่นั้นใช้การได้ไม่เหลือวิสัยที่จะออกแบบแล้ว เราก็อาจเลือกแนวทางออกแบบวิธีใดวิธีหนึ่งได้ ถ้าใช้วิธีสังเคราะห์ก็เท่ากับเราใช้วิธีการที่ตรงไปตรงมา และมีรายละเอียดที่เข้าใจได้พร้อมมูลสร้างแบบที่ตรงกับข้อกำหนดออกมาเลย แต่ในทางปฏิบัตินั้น น้อยครั้งที่จะใช้วิธีสังเคราะห์กับข้อกำหนดแล้วสังเคราะห์แบบออกมาได้สำเร็จในทันทีทันใด หรือถึงแม้จะทำได้แต่เราก็มักจะหาทางป้องกันความผิดพลาดด้วยการตรวจสอบแบบที่ได้กับข้อกำหนดก่อน ถ้าหากไม่ตรงกับข้อกำหนดก็ต้องศึกษาพิจารณาปัญหาแล้วสังเคราะห์แบบใหม่อีกครั้ง ดังแสดงในรูปที่ 1



รูปที่ 1
การออก
วิธีการ

กระบวนการออกแบบที่ใช้กันมากกว่าแบบสังเคราะห์ คือการลองผิดลองถูกควบคู่กับการวิเคราะห์ ซึ่งต้องมีการพิจารณาย้อนกลับไปกลับมา เริ่มต้นด้วยการศึกษาพิจารณาข้อกำหนด แล้วเราก็นึกร่างเค้าโครงแบบขั้นต้น ซึ่งเราคิดว่าน่าจะแก้ปัญหาได้ จากนั้นเราก็วิเคราะห์แบบที่คิดขึ้นนั้นว่ามีส่วนประกอบอะไรบ้าง ทำอะไรได้บ้าง ตรงกับข้อกำหนดที่ตั้งเอาไว้หรือไม่ ถ้าตรงกันก็เป็นอันเสร็จกระบวนการออกแบบ แต่ส่วนใหญ่แล้วแบบที่คิดขึ้นในตอนแรกอาจไม่ตรงกับข้อกำหนดหมดทุกข้อ เมื่อเป็นเช่นนี้เราก็อาจจะต้องออกแบบใหม่ หรือเราอาจศึกษาว่ามีข้อกำหนดอะไรบ้างที่แบบของเราทำงานได้ไม่ตรง ตามนั้น แล้วเราก็เพียงแต่แก้ไขเฉพาะส่วนที่ยังไม่ตรงกับข้อกำหนดเสียใหม่ จากนั้นเราก็เอาแบบที่คิดขึ้นใหม่ไปวิเคราะห์ และตรวจสอบกับข้อกำหนดอีก ทำเช่นนี้เรื่อยๆไป จนกระทั่งเราได้แบบที่ตรงกับข้อกำหนดทุกอย่าง หรือจนกระทั่งเราสรุปว่า ไม่มีทางที่เราจะออกแบบได้ตรงกับข้อกำหนดที่ตั้งเอาไว้ วิธีออกแบบย้อนกลับไปกลับมาเช่นนี้ เรียกว่า เป็นการออกแบบชนิด Iteration (Iterative design) และมีขั้นตอนดังแสดงในรูปที่ 2



ชนิด **Iteration**

วิธีการออกแบบโปรแกรมด้วยผังงาน

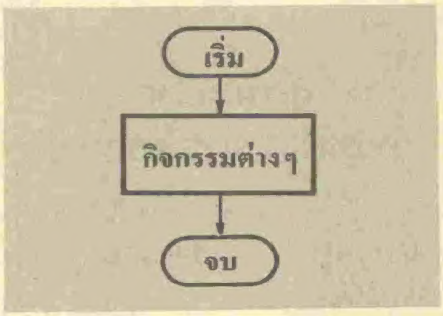
ผู้เขียนโปรแกรมควรอาศัยผังงานช่วยเป็นเครื่องมือในการเขียนโปรแกรม เช่นเดียวกับช่างที่ต้องอาศัยแบบแปลนในการก่อสร้าง กระบวนการออกแบบโปรแกรมด้วยผังงานมีหลายรูปแบบและไม่ถึงกับมีหลักเกณฑ์ที่ตายตัวว่าต้องทำอย่างไร นอกจากแนวความคิดกว้างๆ ว่าควรแบ่งงานออกเป็นกิจกรรมที่เป็นขั้นตอนที่ผู้เขียนโปรแกรมสามารถใช้คำสั่งได้ ต้องมีจุดเริ่มต้นกิจกรรมและจุดที่ทำให้จบกิจกรรมทั้งหมด ตามปรัชญาของกระบวนการออกแบบดั้งเดิมที่นิยมกัน 3 กระบวนการคือ การวิเคราะห์ การสังเคราะห์ และการลองผิดลองถูก ในบทความนี้จะขยายความและประยุกต์ใช้ให้เข้าใจยิ่งขึ้นดังต่อไปนี้

ตัวอย่าง การออกแบบโปรแกรมจัดลำดับค่าของเลข สมมติในที่นี้มีเลขอยู่ 5 จำนวน กำหนดให้ใช้ภาษาเบสิก

ขั้นตอนแรก ทำการวิเคราะห์ คือพิจารณาแจกแจงออกเป็นกิจกรรมต่างๆ จากการทำความเข้าใจให้ถ่องแท้ในเรื่องข้อกำหนดต่างๆที่จะต้องนำมาใส่ให้เครื่องทำงาน ซึ่งต่อไปเรียกว่า อินพุต (input) โดยเฉพาะอินพุตตอนต้น ส่วนใหญ่จะเกี่ยวกับข้อมูล ต้องรู้หรือกำหนดได้ว่าข้อมูลนำเข้าให้เครื่องคอมพิวเตอร์ทำงานลักษณะใด นำเข้าทางใด หรือด้วยคำสั่งแบบใด เช่น อ่านจากแฟ้มที่มีอยู่แล้ว หรือการใส่ข้อมูลจากแป้นพิมพ์ ข้อมูลเมื่อนำเข้าเครื่องแล้วจะเก็บอย่างไร เป็นข้อมูลแบบไหน ที่เก็บมีชื่อว่าอย่างไร เป็นต้น

อีกสิ่งหนึ่งที่ต้องทำความเข้าใจคือสิ่งที่เป็นผลตามความต้องการให้เกิดจากเครื่อง ซึ่งต่อไปเรียกว่า เอาต์พุต (output) โดยเฉพาะเอาต์พุตสุดท้าย ต้องรู้ว่าอะไรเป็นส่วนผลผลิตของการประมวลผล ผลิตเป็นข้อมูลแบบไหน นำแสดงออกหรือจัดเก็บไว้อย่างไรที่ไหน

การทำความเข้าใจกับสิ่งเหล่านี้นับเป็นการเตรียมวิเคราะห์กิจกรรมที่เป็นจุดเริ่มต้น และการหาจุดที่จบของการออกแบบโปรแกรมดังรูปที่ 3



รูปที่ 3 ผังงานแสดงการออกแบบโปรแกรม

เริ่มต้นวิเคราะห์พยายามจำแนกกิจกรรมที่จะนำไปสู่ผลเอาต์พุตสุดท้าย เพราะก่อนที่จะออกเอาต์พุตสุดท้ายที่ต้องการนั้น อาจต้องผ่านหลายกิจกรรม เทคนิคการจำแนกกิจกรรมมี 3 วิธีที่สำคัญ คือ

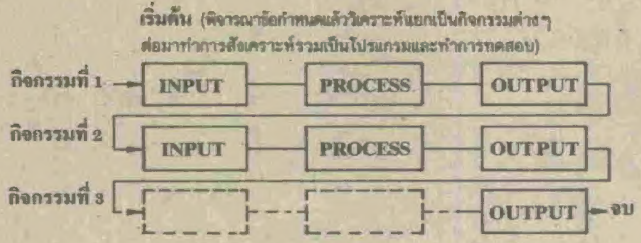
วิธีที่ 1 ไล่กิจกรรมย้อนจากเอาต์พุตสุดท้ายกลับมาจนถึงอินพุตต้น ใช้วิธีการออกแบบชนิดล่างขึ้นบน

วิธีที่ 2 ไล่กิจกรรมจากอินพุตต้นจนถึงเอาต์พุตสุดท้าย ใช้วิธีการออกแบบชนิดบนลงล่าง

วิธีที่ 3 ใช้แบบวิธีแรกผสมกับวิธีที่สอง ใช้วิธีการออกแบบชนิดผสมผสาน

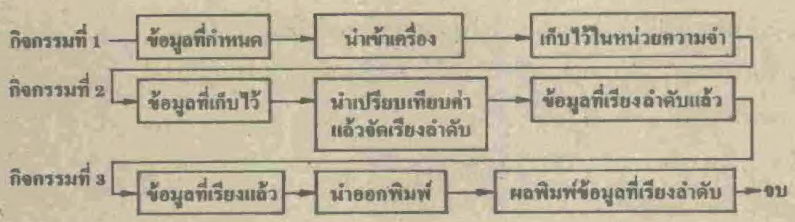
อย่างไรก็ตาม รูปแบบของโปรแกรมทุกชนิดจะมีรูปแบบพื้นฐานเช่นเดียวกันหมดดังรูปที่ 4

4



การวิเคราะห์หากิจกรรมด้วยวิธีต่างๆแล้ว ตามข้อกำหนดให้จัดลำดับค่าเลข 5 จำนวน ในที่นี้จำแนกกิจกรรมได้เป็น 3 กิจกรรมดังรูปที่ 5

5

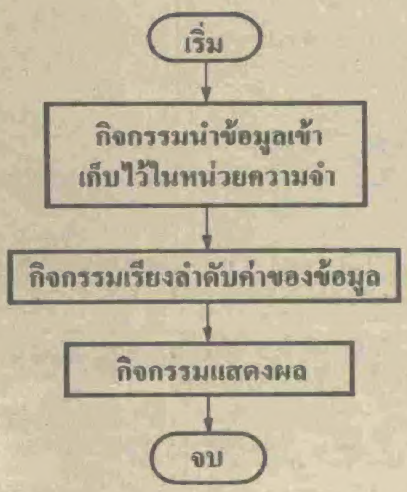


นำผลการวิเคราะห์มาเขียนเป็นผังโปรแกรมจะทำให้เข้าใจขั้นตอนทำงานได้ชัดเจนขึ้นดังรูปที่ 6

ขั้นต่อไปทำการวิเคราะห์ในแต่ละกิจกรรมโดยละเอียด แตกออกเป็นกิจกรรมเล็กๆที่สามารถใช้คำสั่งภาษาเบสิกได้ แล้วค่อยทำการสังเคราะห์คือนำกิจกรรมต่างๆเหล่านั้นมาประกอบกัน หากกิจกรรมใดที่ยากก็พยายามหาตัวอย่างจากโปรแกรมอื่นๆที่มีลักษณะปัญหาใกล้เคียงกันมาเทียบเคียงดู หรือจัดแบ่งไว้เป็นโปรแกรมส่วนย่อยต่างๆ แล้วค่อยกลับมาระดมความคิดเป็นเรื่องเฉพาะที่หลัง การวิเคราะห์ในขั้นที่แตกเป็นโปรแกรมภาษาแต่ละกิจกรรมดังนี้

กิจกรรมที่ 1 นำข้อมูลเข้าเก็บไว้ในหน่วยความจำ ในขั้นตอนนี้ต้องอาศัยความรู้และความชำนาญในการใช้ภาษาเป็นสำคัญ การนำข้อมูลเข้าเครื่องในภาษาเบสิกมีหลายรูปแบบ ในที่นี้เลือกใช้คำสั่ง READ อ่านข้อมูลทั้ง 5 จำนวน โดยนำเข้าทีละหนึ่งจำนวนจนหมด ใช้คำสั่ง FOR...NEXT เข้าช่วย การเก็บค่าทั้ง 5 ไว้ในหน่วยความจำใช้อะเรย์ช่วยเก็บผังงาน และโปรแกรมต้นฉบับสำหรับกิจกรรมนี้ออกแบบได้ดังรูปที่ 7

6




```
10 DIM N(5)                 เตรียมที่เก็บข้อมูลในหน่วยความจำ
20 FOR A = 1 TO 5
30 READ N(A)                กิจกรรมนำข้อมูลเข้าทีละจำนวน
50 DATA 8, 4, 10, 6, 9
40 NEXT A                   เงื่อนไขในการย้อนกลับ
```

7

กิจกรรมที่ 2 เรียงลำดับค่าของข้อมูล โดยการจัดอะเรย์มาเทียบเป็นคู่ เริ่มจากนำค่าในอะเรย์ที่ 1 กับ 2 เป็นต้นไป ทำกิจกรรมเปรียบเทียบจนครบทุกจำนวน แต่ละคู่ค่าที่เทียบกันมีทางเกิดขึ้นได้ 3 ประการ คือประการแรกมีค่าต่างกันค่าในอะเรย์ลำดับที่ต่ำกว่ามีค่าสูงกว่าแล้วก็จะทำให้เกิดอีกกิจกรรมหนึ่ง คือการเลื่อนค่าที่สูงกว่าขึ้นไว้ในอะเรย์ที่ลำดับสูงกว่า ประการที่ 2 ค่าไม่ต่างกันและประการสุดท้ายอะเรย์ลำดับที่ต่ำกว่ามีค่าน้อยกว่า ทั้งสองประการหลังก็ให้เลื่อนไปเทียบกับอะเรย์ที่ลำดับสูงถัดต่อไป เขียนเป็นโปรแกรมต้นฉบับและผังงานสำหรับกิจกรรมนี้ดังรูปที่ 8

```
60 FOR A = 1 TO 4          } จัดการเทียบค่าให้ครบจำนวน ในที่นี้จัดแค่ 4 คู่
70 IF N(A) > N(A+1)        } - ตรวจเงื่อนไข { เทียบอะเรย์ลำดับที่ใกล้กัน ค่าต้นเป็นหลักถ้าค่าในลำดับต้นมีค่าสูงกว่า ก็ไปทำกิจกรรมย่อยแล้วกลับมาใหม่
THEN GOSUB 300             } - ถ้าจริงไปทำกิจกรรมย่อยที่ 300
80 NEXT A                  - เงื่อนไขในการย้อนกลับ
```

8

กิจกรรมย่อย ได้แก่การย้ายค่าที่สูงกว่าของอะเรย์ลำดับต้นเก็บฝากไว้ที่หนึ่ง จัดเลื่อนค่าในอะเรย์ต่อไปมาแทนที่ ต่อจากนั้นจึงย้ายค่าที่ฝากนั้นไปแทนที่อะเรย์ตัวที่เลื่อนค่า ก่อนกลับคืนกิจกรรมหลักทำการเปรียบเทียบอีกครั้งหนึ่ง ก็ทำการลดลำดับของอะเรย์ลงหนึ่งคู่ ผังงานและโปรแกรมต้นฉบับเขียนได้ดังรูปที่ 9

```
300 LET X=N(A)             - ฝากค่าอะเรย์ลำดับต้นไว้ใน X
310 LET N(A)=N(A+1)        - เลื่อนค่าอะเรย์ถัดไปมาไว้ที่อะเรย์ลำดับต่ำ
320 LET N(A+1)=X           - ย้ายค่าที่ฝากนำมาเก็บไว้ในอะเรย์ลำดับถัดไป
330 LET A = A-2            - ลดอะเรย์ลง 1 คู่
350 RETURN                 - กลับในกิจกรรม
```

9

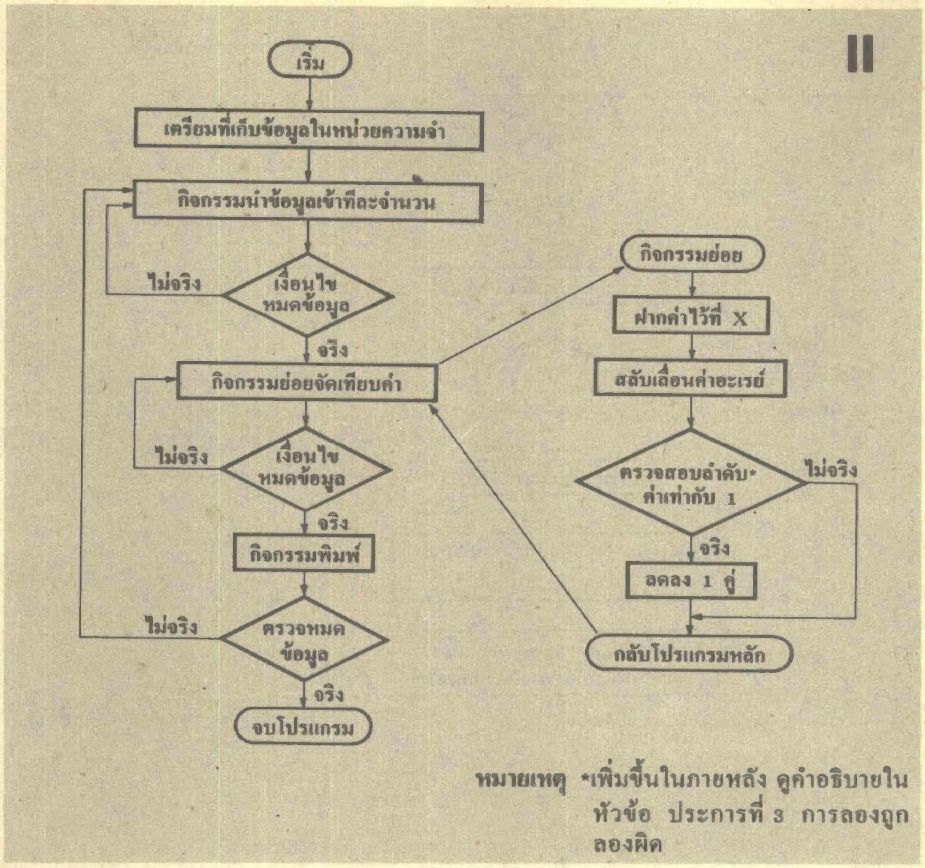
กิจกรรมที่ 3 การแสดงผลการเรียงลำดับค่าของข้อมูลอินพุต โดยเรียงจากน้อยไปมากแสดงออกด้วยการพิมพ์ข้อมูลที่เรียงแล้ว ผลพิมพ์ที่ต้องการตามตัวอย่างนี้คือ

OUTPUT
4, 6, 8, 9, 10

วิเคราะห์นำข้อมูลที่จัดเรียบร้อยแล้วออกพิมพ์ โดยทยอยออกตามลำดับอะเรย์ ผังงานและโปรแกรมต้นฉบับเขียนได้ดังรูปที่ 10

```
100 LPRINT "OUTPUT"        พิมพ์คำ "OUTPUT"
110 FOR A=1 TO 5           แสดงผลจนหมดข้อมูล
120 LPRINT N(A);";";       พิมพ์ผลข้อมูลทีละจำนวน
130 NEXT A                 ตรวจเงื่อนไขในการย้อนกลับ
140 END                    จบโปรแกรม
```

10



11

หมายเหตุ *เพิ่มขึ้นในภายหลัง ดูคำอธิบายในหัวข้อ ประการที่ 3 การลองถูกลองผิด

ต่อจากนี้เป็นการตกแต่งให้โปรแกรมกระชับหรือทำงานได้รวดเร็วขึ้น เช่น การทำหมายเหตุอธิบาย หรือปรับแต่งภาษาให้มีประสิทธิภาพ การทำเช่นนี้ได้ดีมากน้อยขึ้นอยู่กับประสบการณ์และความชำนาญในการใช้ภาษานั้นๆ ตามตัวอย่างนี้สามารถปรับแต่งการใช้ภาษาให้กระชับขึ้นในกิจกรรมสลับค่าของอะเรย์ตามประโยคคำสั่งที่ 300 ถึง 320 ด้วยการใช้คำสั่ง SWAP แทนดังรูปที่ 12

ขั้นตอนที่สอง การสังเคราะห์ด้วยการนำกิจกรรมต่างๆประกอบขึ้นเป็นผังงานทั้งหมดดังรูปที่ 11

ขั้นตอนที่สาม การลองถูกลองผิดด้วยการทดสอบแล้วปรับแต่งตามตัวอย่างจะพบว่าเมื่อลองทดสอบแล้วยังมีข้อบกพร่องคือ เกิดกรณีที่กิจกรรมย่อยตอนลดลำดับอะเรย์ลงหนึ่งคู่หากลำดับของอะเรย์เท่ากับ 1 เมื่อลบด้วย 2 ลำดับค่าของอะเรย์จะติดลบทันที ซึ่งภาษาเบสิกถือว่าเป็นการผิดจะหยุดโปรแกรม จึงต้องเพิ่มกิจกรรมทำการตรวจสอบไม่ให้ลำดับอะเรย์เท่ากับค่า 1

```
325 IF A=1 THEN 350   ตรวจไม่ให้ค่าเท่ากับ 1
                      ถ้าเท่าให้ข้ามไปจบกิจกรรม
```

โปรแกรมต้นฉบับเดิม	โปรแกรมต้นฉบับใหม่
300 LET X=N(A)	
310 LET N(A)=N(A+1)	300 SWAP N(A), N(A+1)
320 LET N(A+1)=X	

ดังนั้นโปรแกรมที่หลังตกแต่งแล้วควรจะเป็นคล้ายดังต่อไปนี้

```
  5 'START PROGRAM
 10 DIM N(5)
 20 FOR A=1 TO 5
 30 READ N(A)
 40 NEXT
 50 DATA 8, 4, 10, 6, 9
 60 FOR A=1 TO 4
 70 IF N(A) > N(A+1)
    THEN GUSUB 300
 80 NEXT
100 LPRINT "OUTPUT"
110 FOR A=1 TO 5
120 LPRINT N(A);",";
130 NEXT
140 END
300 SWAP N(A), N(A+1)
325 IF A=1 THEN 350
330 A=A-2
350 RETURN
```

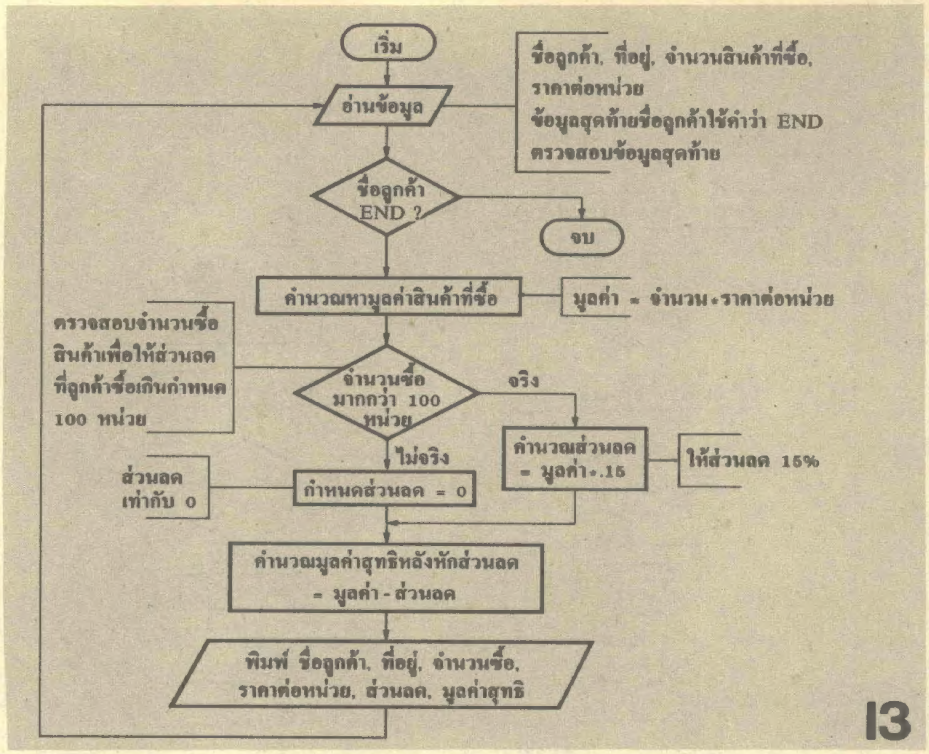
12

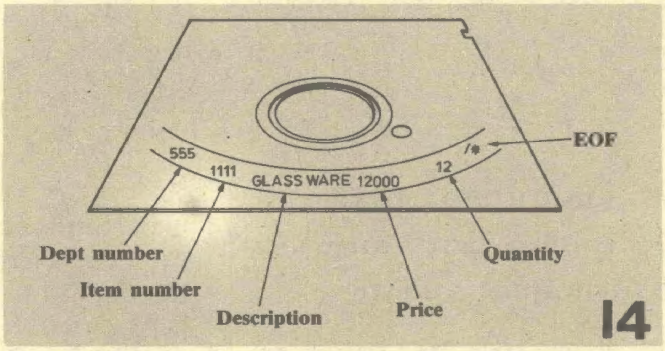
ลองเปรียบเทียบระหว่างผังงานกับโปรแกรมต้นฉบับ พิจารณาการทำความเข้าใจของขั้นตอนทำงานของโปรแกรม จะเห็นว่าผังงานเข้าใจได้ง่ายกว่าโปรแกรมต้นฉบับ

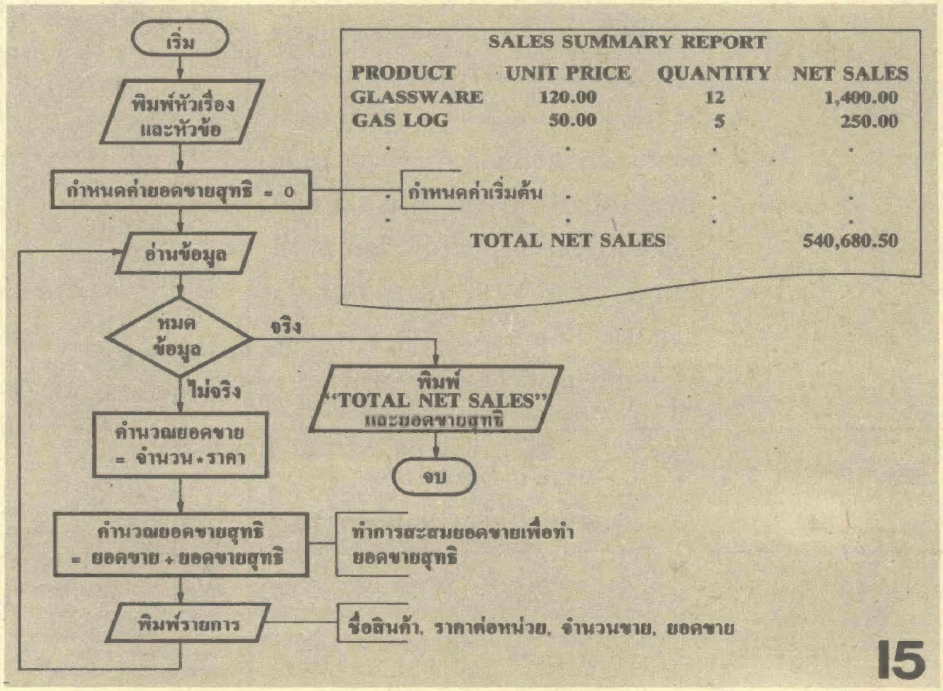
กระบวนการออกแบบผังงานทั้ง 3 วิธีบางครั้งก็ใช้ร่วมกัน แต่การสังเคราะห์และการลองผิดลองถูกนั้นดูผิวเผินแล้วรู้สึกว่าไม่ค่อยมีหลักการ เพราะไม่ได้บอกว่าต้องนำอะไรมาประกอบหรือนำอะไรมาลองผิดลองถูก แต่กระบวนความคิดนี้ได้อาศัยรูปแบบที่มีผู้คิดค้นหาการประมวลผลแบบต่างๆ สะสมไว้นำมาเทียบเคียง รูปแบบการประมวลผลที่สำคัญได้แก่ การเลือก การคำนวณ การพิมพ์ การนำข้อมูลเข้า การเรียงข้อมูล การใช้ตารางข้อมูล การเพิ่มปรับปรุงข้อมูลให้ทันสมัย การทำยอดรวมย่อย จะยกตัวอย่างบางรูปแบบดังต่อไปนี้

รูปแบบที่ 1 การอ่านข้อมูลและการเลือก ตัวอย่างผังงานการให้ส่วนลดราคากับลูกค้าที่ซื้อสินค้าเกินจำนวนที่กำหนดดังรูปที่ 13

รูปแบบที่ 2 การพิมพ์และการคำนวณ ตัวอย่างการทำรายงานการขายสินค้า ดังรูปที่ 14 และ 15

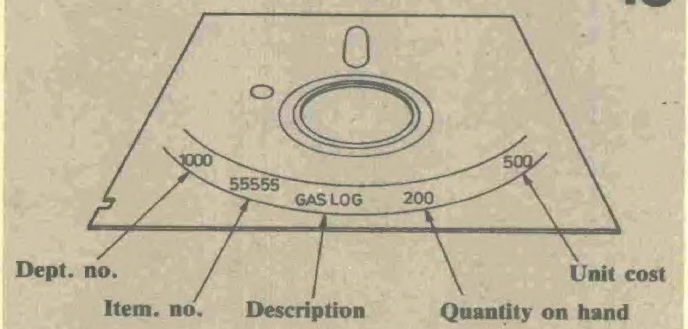






รูปแบบที่ 3 การทำยอดรวมย่อย (control break) ตัวอย่างข้อมูลสินค้าคงคลัง ต้องการให้ทำรายงานสินค้าคงคลังของทุกแผนกและให้ทำยอดรวมย่อยแบ่งเป็นแผนกดังรูปที่ 16, 17 และ 18

16

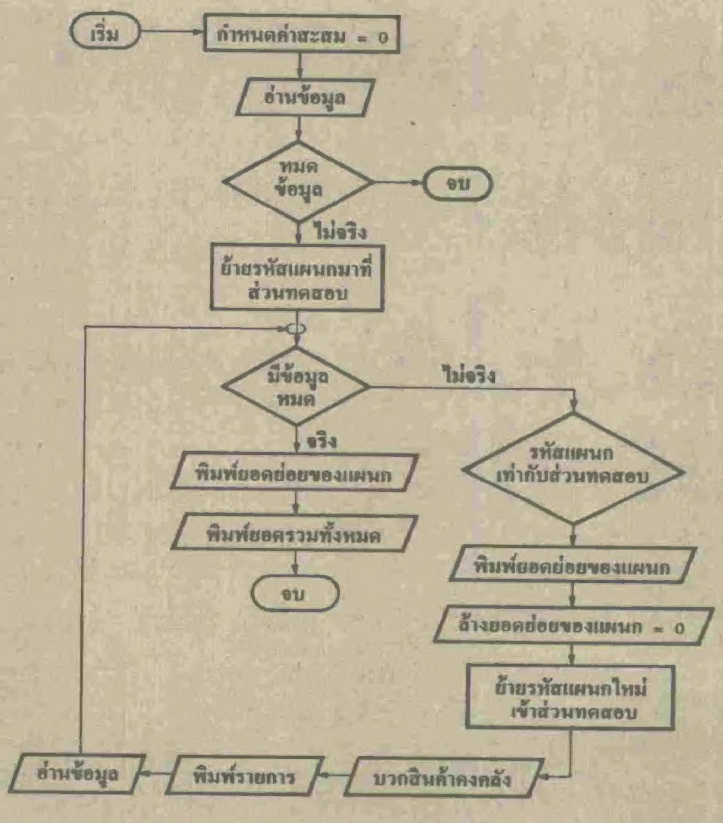


17

INVENTORY REPORT PAGE 1

DEPT NO	ITEM NUMBER	QUANTITY ON HAND	UNIT COST	IVENTORY VALUE
1000	55555	200	5.00	1,000.00
	33333	500	10.50	5,250.00
	TOTAL VALUE OF DEPT 1000			6,250.00*
2000	11111	100	10.00	1,000.00
	TOTAL VALUE OF DEPT 2000			1,000.00*
4000	22222	400	4.00	1,600.00
	33333	600	3.00	1,800,00
	55555	50	5.00	250.00
	TOTAL VALUE OF DEPT 4000			3,650.00*
	TOTAL TOTAL IVENTORY VALUE			10,900.00**

18



ข้อสรุป

การออกแบบโปรแกรมด้วยผังงานเริ่มต้นจากข้อกำหนดที่ระบุว่าต้องการให้ทำอะไรและได้ผลที่ต้องการเป็นอย่างไร เทคนิคที่ใช้ในการออกแบบผังงานมีหลายวิธี ที่ยกตัวอย่างเฉพาะที่นิยมกันคือการวิเคราะห์ การสังเคราะห์ และการลองผิดลองถูก ทั้งนี้อาศัยประสบการณ์เป็นหลักสำคัญ ดังนั้นจึงมีการประมวลรูปแบบหรือแนวความคิดต่างๆหลายรูปแบบไว้ให้เลือกใช้ที่นำไปวิเคราะห์หรือการสังเคราะห์ เทคนิคการออกแบบผังงานยังมีอีกหลายวิธีโดยเฉพาะเทคนิคโครงสร้างที่มีผู้นิยมใช้มากในปัจจุบันจะได้กล่าวถึงรายละเอียดในครั้งต่อไป